IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Patent Application No. 09/449,912

Applicant: Divittorio

Filed: December 2, 1999

TC/AU: 2195

Examiner: Kenneth Tang

Docket No.: 202232 (Client Reference No. 99,032 US)

Customer No.: 23460

APPELLANT'S REPLY UNDER 37 C.F.R. SECTION 41.41

Mail Stop Appeal Brief – Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

This paper is filed in response to the Examiner's Answer mailed on November 16, 2007. Appellant hereby respectfully requests allowance of the pending claims for the reasons set forth in Appellant's Appeal Brief filed on September 6, 2007, and for the further reasons stated herein.

Status of Claims

Claims 1-26 stand rejected, and the rejection of these claims is presently being appealed.

For convenience, a complete listing of these claims appears in the Claims Appendix.

Grounds of Rejection to be reviewed on Appeal

The grounds of rejection to be reviewed on appeal are the grounds stated in the Office Action mailed on October 6, 2006. In particular, Appellant appeals the Office Action's:

1. Rejection of Claims 1-7, 13-19, and 25-26 as obvious under 35 U.S.C. Section 103(a) over Applicant's Admitted Prior Art (AAPA) in view of Sinibaldi et al., U.S. Patent No. 6,549,945 (Sinibaldi); and

2. Rejection of Claims 8-12 and 20-24 as obvious under 35 U.S.C. Section 103(a) over AAPA in view of Sinibaldi and Messih et al. U.S. Patent No. 5,526,794 (Messih).

Argument in Reply to Examiner's Answer

Appellant has carefully reviewed the Answer dated November 16, 2007. Section (9) "Grounds of Rejection" of the Answer, beginning at page 3 and ending at page 7, is virtually identical (a reference to col. 13, lines 20-34 of Messih was added to paragraph 13 on page 6) to the grounds recited in the Office Action from which the present appeal was taken. The citation to Messih's disclosure is, however, irrelevant to claim 8-12 since claims 8-12 call for the lower priority "embedded control task" to periodically repeat according to a specified "repetition cycle parameter" which is neither disclosed nor suggested in Messih. Appellant has addressed the Answer's grounds for rejection in the previously filed Appellant's Appeal Brief, and thus Appellant will not repeat these other arguments.

Appellant seeks reversal of the rejection of claims 1-26 (provided in the Claims Appendix attached hereto) that are directed to a control processor that, in addition to executing a set of control blocks, executes an embedded control task comprising a *multivariable linear program* including a set of outputs corresponding to process set points (provided to the control blocks). The claimed invention specifies a particular multiple program execution scheme wherein the embedded control task executes at a lower execution priority than an execution priority for the set of control blocks. The recited specific program execution scheme is not obvious over the prior art, including AAPA, Sinibaldi, and Messih.

The Answer, and the references cited therein, fails to provide any reasonable basis for one skilled in the art at the time of the invention to even explore the possibility of developing Appellant's process control program execution scheme. More particularly, increasing the processing load on a control processor increases the processor's computational burden. Therefore, contrary to the Office Action's assertion, the claimed system becomes less "flexible" (due to the enhanced processing load placed on the control processor) and therefore the claimed processing arrangement would not be obvious to one skilled in the art in view of the cited teachings of the prior art (including col. 1, lines 50-55 of Sinibaldi).

Appellant's Responses To Section (10) of the Answer

1. The Rejection of Claims 1, 6, 7, 13, 18, 19, 25 and 26

The continued rejection of Appellant's claim 1 is premised upon multiple fundamental errors in the Office Action and subsequently issued Answer. Contrary to the assertions contained in the Answer, Appellant submits: (1) Sinibaldi neither discloses nor suggests a "multivariable linear program", and (2) increased "flexibility", the stated basis (at page 4, lines 10-12 of the Answer) for one skilled in the art to incorporate the specifically recited program execution scheme in a control processor, has not been shown to result from modifying the prior art control processor in accordance with the claimed invention.

Addressing the first point raised by Appellant with regard to the Answer's "Response to Argument", Sinibaldi does not disclose a "multivariable linear program" for computing process control variable setpoints. Appellant respectfully submits that, contrary to the Answer's assertion in the last paragraph of page 7, Sinibaldi neither discloses nor suggests the "multivariable linear program" recited in independent claims 1 and 25.

The entry for "linear program" in Wikipedia, for example, supports Appellant's definition of a linear program. In particular, the Wikipedia entry states:

> "In mathematics, linear programming (LP) problems involve the optimization of a linear objective function, subject to linear equality and inequality constraints. Put very informally, LP is about trying to get the best outcome (e.g. maximum profit, least effort, etc) given some list of constraints (e.g. only working 30 hours a week, not doing anything illegal, etc), using a linear mathematical model."

Appellant respectfully submits that the "matrix" described at col. 18, lines 25-28 of Sinibaldi does not constitute or imply the presence of a linear program in Sinibaldi's disclosed system. The "matrix" described at column 18, lines 25-28 is a table that merely lists a set of tasks performed by a digital signal processor (DSP) and the associated computing (MIP) requirements for each task. Appellant submits that the above-provided Wikipedia definition, consistent with Appellant's own description of a linear program, satisfactorily rebuts the Answer's assertion that Sinibaldi discloses a multivariable linear program at column 18, lines 25-28, and FIG. 14. There is no reasonable way to read Sinibaldi's "matrix" (a listing of tasks and associated computing requirements) on Appellant's claimed multivariable linear program. Thus, Sinibaldi can not disclose a multivariable linear program operating at a lower execution priority than an execution priority of a set of control blocks, and the premise upon which the Office Action's rejection of claim 1 (and claim 25) relies is unsupported by the actual teachings of Sinibaldi. In the event that the rejection of claims 1 and 25 is not withdrawn, **Appellant requests identification of a reference that defines a "linear program" in a manner that supports the Answer's broad interpretation that apparently includes any table/matrix.**

Addressing the second point raised above with regard to the Answer's "Response to Argument," the Answer has not provided any support for its assertion that combining Sinibaldi's matrix with the admitted prior art would "increase flexibility" with regard to the execution of tasks on the control processor. The Answer cites col. 1, lines 50-55 of Sinibaldi. Appellant notes that the referenced portion of Sinibaldi refers to the "interfaces of DSP Adaptors" and does not appear to relate to execution of control tasks on a control processor – to which Appellant's claimed invention is directed. In fact, if anything, the claimed invention reduces the flexibility of the control processor since it imposes additional processing load on the control processor that it may not be able to handle. More specifically, executing the linear program at a lower priority on the control processor introduces a risk that insufficient CPU time will be allocated to complete the linear program processing (due to the processor load of the higher priority cyclically executed control blocks). Thus executing the linear program at a lower priority than block processing on a control processor could, for example, limit the number of control blocks that are handled by the control processor. Alternatively, the minimum repetition period for processing the control blocks may need to be increased to ensure sufficient time is available between block processing cycles to complete the linear program. Thus, contrary to the Answer's assertion at the bottom of page 7, the claimed invention reduces flexibility.

Regarding the more particular arguments set forth on pages 8-15 of the Answer, Appellant notes that the statements primarily repeat the points presented in the general comments that begin at page 7 of the Answer. Those points, "matrix/linear program" and "increased flexibility" are addressed above and will not be repeated here. The remaining points are addressed herein below. Throughout the repeated discussion, the Answer does not establish a teaching in the prior art to render Appellant's claimed linear program (low priority) and control block (high priority) execution scheme on a control processor. Both of the claimed executables are essential to the proper operation of a system, and there are substantial disincentives (including *reduced* design flexibility) for executing the executables on a single processor.

At page 8 the Answer erroneously states that Sinibaldi teaches a matrix wherein the entries of the matrix are "setpoints." Appellant submits that Sinibaldi does not describe the contents of the matrix as being "setpoints". Rather than disclosing Appellant's claimed "process setpoints", Sinibaldi's matrix contains a list of tasks and the MIPs requirements for each task. Appellant submits that a MIPs requirement does not constitute the claimed process setpoint.

At page 10, lines 4-10, the Answer discusses "boosting" a program's priority when it is time to execute. Appellant respectfully submits that the invention is directed to assigning relative priorities to executables (e.g., multivariable linear program providing setpoints and a set of control blocks) to ensure that the control block is given preference over the linear program. Thus, there is no "boosting" or changing of a priority based upon a need to complete the executable.

At page 10, last paragraph, the Answer seeks to justify the assertion that "increased flexibility" results from adopting a priority-based scheme of Sinibaldi. However, the Answer has still not demonstrated how incorporating Sinibaldi's teachings to applicant's admitted prior art (AAPA) in a way that renders Appellant's claimed invention somehow enhances the flexibility of the system. Appellant has previously explained that the claimed invention actually reduces the flexibility of the control processor's operation and thus constitutes a disincentive – rather than an incentive – to apply Sinibaldi's disclosure to AAPA. Thus, Appellant's claimed control processor having the recited execution priority scheme is not obvious over AAPA in view of Sinibaldi.

The discussion beginning at page 11 of the Answer is indeed repetitive of previous discussions. Appellant respectfully submits that the Answer is applying an unduly broad interpretation of the term "linear program" which is unsupported by the well known definition for "linear program."

The discussion beginning at page 12 of the Answer is indeed repetitive of previous discussions. The Answer merely repeats a fundamental shortcoming of its obviousness argument. Sinibaldi does not disclose executing a multivariable linear program at a relatively low priority in comparison to a set of control blocks. In fact, Sinibaldi does not disclose a multivariable linear program. The conclusion of the section, on page 13, assumes, without any basis in fact, that the output of the multivariable linear program is less important than the output of the control blocks.

The discussion beginning at page 13 of the Answer is also repetitive of previous discussions. See, e.g., the discussion beginning at page 12.

The discussion beginning at page 14 of the Answer states that claim 1 does not recite "calculation of set point values." Though claim 1 does not use the specific words "calculation of set point values," claim 1 recites the "output" of the claimed "multivariable linear program" includes values "corresponding to process setpoints." This does not detract from Appellant's primary point that Sinibaldi does not disclose a "multivariable linear program." The remainder of the discussion is merely repetitive of previous discussion.

The above discussion focused primarily upon claims 1 and 25. With regard to the remaining independent claims 13 and 26, Appellant notes that while the term "linear program" is not used, each of these claims recites a program component that executes at a relatively low priority on a control processor and generates setpoint values for a process. Claim 26 recites additional features relating to cyclic execution of a lower priority sequence of instructions that provide process control set points.

2. The Rejection of Claims 2-3, 5, 14 and 17

Appellant notes that the reference to a "multi-level" execution scheme refers to the claimed multiple, distinct execution priorities. The Answer has still not identified any teaching in the prior art of executing "supervisory control blocks" as a part of the set of control blocks that are executed at a relatively higher priority than the linear program. Appellant notes that the Answer does not address the specific request to identify the basis upon which each of the rejections relies. See, e.g., the recited elements of claims 2 and 3 identifying a supervisory control block and a particular type of function carried out by the supervisory control block.

3. The Rejection of Claims 4 and 16

The Answer has not identified the claimed multivariable control block having the recited functionality. The references to AAPA and Sinibaldi in the Answer are irrelevant to the claimed invention since neither discloses carrying out the claimed function directed to transferring a "process control model" implemented by the claimed "embedded control task" on the control processor.

4. The Rejection of the Claim 15

Appellant notes that the Answer has identified AAPA and Sinibaldi as disclosing the elements of claim 15. However, neither AAPA nor Sinibaldi even remotely discloses the claimed multivariable control block and the recited step of downloading data from a workstation to a database accessed by the multivariable control block.

5. The Rejection of Claims 8, 11, 20 and 23

Appellant reiterates the earlier argument that none of the cited prior art references discloses a repetition cycle parameter specifying a period for re-commencing a cycle of the *embedded control task.* The Answer's references to Messih and AAPA are irrelevant since neither discloses a repetition period being specified for the lower priority embedded control task/multivariable control application (corresponding to a "background" process in the context of Messih).

6. The Rejection of Claims 9 and 21

The prior art does not even remotely disclose or suggest the claimed control processor arrangement wherein one of the supervisory control blocks of the set of control blocks (executing at the higher priority level) determines when the claimed repetition period of the embedded control task/control application has expired. Appellant cannot determine the relevance of citations provided in the Answer.

7. The Rejection of Claims 12 and 24

The Answer's citation to col. 3, lines 20-34 of Messih does not appear to relate to the recited claim elements.

Conclusion

In summary, the claimed invention is directed to a particular execution scheme for a control processor. The control processor executes both a set of control blocks as well as an embedded control task providing setpoints for a process. The embedded task executes at a relatively lower priority than the set of control blocks. The present invention is not rendered obvious by the combined teachings of AAPA, Sinibaldi, and Messih. Contrary to the Answer's assertions, the Sinibaldi reference does not disclose a multivariable linear program or an embedded multivariable control application that provides setpoints for a process. Furthermore, incorporating a multivariable linear program into a control processor would have increased the processor load and introduced limitations during normal operation of the control processor that do not exist in prior art systems wherein supervisory applications run on separate workstations. Thus, the present invention actually reduces control processor flexibility. One skilled in the art would therefore not modify AAPA in view of Sinibaldi and Messih to render Appellant's claimed invention. For these reasons, as well as others stated herein above and in the Appeal Brief, the presently pending claims are patentable over the prior art presently known to Appellant.

Appellant therefore requests reversal of the presently pending rejection of claims 1-26.

Respectfully submitted,

Mark Joy, Reg. No. 35,562
LEYDIG, VOIT & MAYER, LTD.
Two Prudential Plaza
180 North Stetson Ave., Suite 4900
Chicago, Illinois 60601-6731
(312) 616-5600 (telephone)
(312) 616-5700 (facsimile)

Date: January 16, 2008

Claims Appendix

1. (Previously presented) A control processor for executing a set of control tasks defining dynamic model-based interactive control of an industrial process, the control processor comprising:

an embedded control task comprising a multivariable linear program including a set of output values corresponding to process setpoints; and

a set of control blocks including regulatory control blocks having output values that are transmitted by the control processor to field devices coupled to the industrial process, wherein the embedded control task executes at a lower execution priority than an execution priority of the set of control blocks.

2. (Original) The control processor of claim 1 wherein the set of control blocks comprise supervisory control blocks.

3. (Original) The control processor of claim 2 wherein the supervisory control blocks include a multivariable control block including computer instructions facilitating communication of data between the control processor and a workstation.

4. (Original) The control processor of claim 3 wherein the multivariable control block includes computer instructions for receiving and storing a process control model to be implemented by the embedded control task.

5. (Original) The control processor of claim 2 wherein the supervisory control blocks include at least one multivariable loop block including computer instructions for providing an input value for a regulatory control block.

6. (Previously presented) The control processor of claim 5 wherein the regulatory control block is a proportional-integral-derivative block.

7. (Original) The control processor of claim 5 wherein the regulatory control block is a ratio block.

8. (Previously presented) The control processor of claim 1 further comprising a repetition cycle parameter specifying a period for re-commencing a cycle of the embedded control task.

9. (Original) The control processor of claim 8 wherein the set of control blocks includes a supervisory control block including a sequence of instructions to determine when to re-commence a cycle of the embedded task in accordance with a value specified by the repetition cycle parameter.

10. (Original) The control processor of claim 1 further comprising a block processing cycle parameter specifying a repetition period for re-commencing a cycle of executing the set of control blocks.

11. (Original) The control processor of claim 10 further comprising a repetition cycle parameter specifying a period for re-commencing a cycle of executing the embedded control task.

12. (Original) The control processor of claim 11 wherein a period specified by the repetition cycle parameter exceeds a period specified by the block processing cycle parameter.

13. (Previously presented) A method for operating a control processor, in an industrial process control environment, to establish operating values including a set of setpoint values and a set of process control variables associated with control elements in a controlled industrial process based upon a set of input variables including process variables provided to the control processor and representing the present state of the controlled industrial process, the method comprising the steps of:

executing, by the control processor, an embedded multivariable control application including computer instructions facilitating computing a setpoint value corresponding to a process control variable; and

executing, by the control processor, a set of control blocks including regulatory control blocks for receiving and storing a set of process variables representing the present state of a controlled process, wherein the embedded multivariable control application executes at a lower execution priority than an execution priority of the set of control blocks.

14. (Original) The method of claim 13 wherein the set of control blocks comprise supervisory control blocks.

15. (Original) The method of claim 14 wherein the supervisory control blocks include a multivariable control block and further including the step of downloading data from a workstation to a database accessed by the multivariable control block.

16. (Original) The method of claim 15 further comprising the steps of receiving and storing, within the database accessed by the multivariable control block, a process control model to be implemented by the embedded multivariable control application.

17. (Original) The method of claim 14 wherein the supervisory control blocks include at least one multivariable loop block, and further comprising the step of providing an input value for a regulatory control block in accordance with execution of instructions and data associated with the at least one multivariable loop block.

18. (Previously presented) The method of claim 17 wherein the regulatory control block is a proportional-integral-derivative block.

19. (Original) The method of claim 17 wherein the regulatory control block is a ratio block.

20. (Previously presented) The method of claim 13 further comprising the step of maintaining a repetition cycle parameter specifying a period for re-commencing a cycle of the embedded multivariable control application.

21. (Original) The method of claim 20 wherein the set of control blocks includes a supervisory control block, and further comprising the step of determining, by the supervisory control block, when to re-commence a cycle of the embedded multivariable control application in accordance with a value specified by the repetition cycle parameter.

22. (Original) The method of claim 13 further comprising the step of maintaining a block processing cycle parameter specifying a repetition period for re-commencing a cycle of executing the set of control blocks.

23. (Previously presented) The method of claim 22 further comprising the step of maintaining a repetition cycle parameter specifying a period for re-commencing a cycle of executing the embedded multivariable control application.

24. (Original) The method of claim 23 wherein a period specified by the repetition cycle parameter exceeds a period specified by the block processing cycle parameter.

25. (Previously presented) An industrial process control computer having multiple operating levels including:

a background control program execution level wherein the process control computer executes an embedded multivariable process control application, the embedded control application including instructions for executing a multivariable linear program to generate a set of values corresponding to process control variable setpoints; and

a foreground control block execution level wherein the process control computer executes a set of control blocks, at a higher execution priority level than the background control program execution level, the set of control blocks including program instructions that, when executed, receive and store a set of process variable values representing the state of a controlled process.

26. (Previously presented) A multi-level multivariable industrial process control program execution framework for an industrial control processor including:

a first cyclically executed sequence of instructions, repeatedly executed according to a first configurable repetition period and at a first level of execution priority, the first cyclically executed sequence of instructions including at least a set of instructions for calculating a setpoint value for a process control variable; and

a second cyclically executed sequence of instructions, repeatedly executed according to a second repetition period and at a second level of execution priority, the second level of execution priority exceeding the first level of execution priority, and thus enabling the control processor to temporarily suspend execution of the first cyclically executed sequence of instructions in order to execute the second cyclically executed sequence of instructions.